

2020-03-05

Multipath Protections and Dynamic Link Recovery in Softwarized 5G Networks using Segment Routing

Barakabitze, Alcardo Alex

<http://hdl.handle.net/10026.1/15231>

10.1109/GCWkshps45667.2019.9024556

2019 IEEE Globecom Workshops (GC Wkshps)

IEEE

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

Multipath Protections and Dynamic Link Recovery in Softwarized 5G Networks using Segment Routing

Alcardo Alex Barakabitze, Lingfen Sun, Is-Haka Mkwawa and Emmanuel Ifeachor

School of Computing, Electronics and Mathematics, Plymouth University, UK

E-mail: {alcardoalex.barakabitze, L.Sun, is-haka.mkwawa, E.Ifeachor}@plymouth.ac.uk

Abstract—Future softwarized 5G networks have to be robust enough so as to ensure high network reliability and services availability. 5G network architecture should make sure that any failed parts in the network are detected, restored and recovered within a permissible period of time and at the lowest achievable cost. In this paper, we propose a Multipath Protection and Link-Failure free MPTCP/SR-based SDN/NFV architecture that increases survivability, resilience, availability of services in 5G networks. We present our system model and a multiPath protection and dynamic Link- Failure free algorithm called "*PathReLief*" that greatly reduces the failure recovery time and avoids link congestion in MPTCP/SR SDN/NFV 5G networks. To demonstrate the effectiveness of our proposal, we compare the performance of the proposed algorithm and the conventional topology discovery mechanisms for link/node failures in POX and OpenDaylight controllers. Preliminary results show that, our approach outperforms others used in the commonly used controllers (i.e., POX and OpenDaylight), in terms of reduced failure recovery time and localization time.

Keywords—SDN, NFV, Segment Routing (SR), MultiPath TCP (MPTCP), link recovery, 5G, path protections

I. INTRODUCTION

5G networks are expected to be 10 - 100 times faster than 4G/LTE networks, support and provide approximately 100% services availability [1] and transmission/network reliability of 99.999%. 5G is set to provide access to any service with better quality to end-users at anytime, anywhere through reliable and cost effective communications, over any medium and across multi-operator domains using different technologies such as SDN and NFV. Requirements such as high throughput, resilience and reliability, consistency and service availability, ultra-low-latency have to be achieved, for 5G to support applications such as live video streaming, AR/VR and video gaming [1]. To cope with these multimedia services which require optimal throughput, Multi-Path Transmission Control Protocol (MPTCP) [2] and advanced forwarding technologies such as Segment Routing (SR) [3] have been proposed. MPTCP is now an integral part of 5G mobile network architecture as a standard feature of 3GPP Release 16. As shown in our previous works [4], [5], MPTCP and SR implementation can be integrated with SDN and NFV to achieve better network resources utilization that leads to higher network throughput and the end-user's QoE.

However, the performance of future softwarized 5G infrastructures can be affected by link or node failures which are

inevitable occurrences in the networks. In SDN/NFV-based network, not only the centralized controller itself can be a single point of failure but also the data plane. If the SDN controller or link/node fails, then the routing and forwarding capabilities of the 5G network become down and can lead to undelivered data packets, unreliable network and even drop new flow requests from end-users. To overcome this, future 5G softwarized networks should be robust enough to ensure high reliability and availability of services by making sure that any failed parts in the network are detected, restored and recovered within a permissible period of time and at the lowest achievable cost.

In this paper, we extend our previous works [4], [5] where it was demonstrated that, MPCTP and SR can improve system throughput while achieving the maximum end-user's QoE in SDN/NFV networks. We provide new approaches for path protections and dynamic link recovery mechanism in MPTCP/SR-enabled SDN/NFV-based 5G networks. The contributions of this paper are three fold:

- We propose the path protections and Link- Failure free MPTCP/SR-based architecture that increases survivability, resilience, availability and robustness of 5G networks.
- We present our system model and a Multipath Protections and Dynamic Link- Failure free algorithm called "*PathReLief*" that greatly reduces failure recovery time and avoids link congestion in MPTCP/SR SDN/NFV 5G networks.
- We perform extensive experiments and evaluate the performance of the proposed PathReLief algorithm over the developed softwarized network platform.

The rest of this paper is organized as follows: Section II presents related work. Section III provides an overview of path protections using segment routing while section IV presents the proposed Multipath Protections and Link- Failure free MPTCP SDN/NFV SR-based architecture. We formulate the multipath protection and dynamic link- failure free SDN/NFV system model and present the proposed PathReLief algorithm in section V. Section VI presents the performance and evaluation of the proposed SDN/NFV system. Finally, section VII concludes the paper.

II. RELATED WORK

Path restoration and protection are the most commonly used failover approaches in SDN and NFV [6]. For path protection strategies [7], [8], [9], backup paths are proactively (pre-installed) configured. That way, backup paths are activated

automatically in the data plane switches when a link fails. Sgambelluri *et.al* [7] propose an OpenFlow architecture that ensures network resource utilization are effectively utilized when the failed link is recovered in Ethernet-based networks. Cheng *et.al* [8] propose a congestion-aware local fast reroute (CALFR) in SDN, a resilience approach that balance failure recovery time and forwarding rule occupation. Authors in [9] introduce a fast failover strategy that reduces the recovery time using preconfigured primary and secondary paths which are calculated by an OpenFlow controller. There are two drawbacks with the path protection mechanisms. Firstly, the preconfigured flow rules in an SDN switch are stored in the Ternary Content Addressable Memory (TCAM), which is expensive, power hungry and limited in size (e.g., can only hold 1500 forwarding rules) [10]. Secondly, the backup paths are correlated and need to be reassigned making this approach static and rigid [8].

With path restoration approaches [6], [11], backup paths are reactively installed when the link/node failure is detected by the SDN controller in the network. However, the failure recovery time using path restoration is larger because the SDN controller has to (a) detect the failed link/node, (b) recompute a new routing path, and (c) install flow entries in the OpenFlow switches directly connected to the failed link. For example, Xiong *et.al* [11] propose a Fast Restoration with Triggered Precomputation (FR-TP) approach that achieves fast failure recovery with minimal resource overhead. The path restoration process which involves the SDN controller and OpenFlow switches, would lead to an increased failure recovery time and a large number of flow rules in SDN switches. Again, for video streaming services, congestion can occur in the network because of redirecting large amount of video flows on the failed link to a new reroute path [8]. Previous studies have shown that the path restoration can take a recovery time of over 100 ms, which is longer than 50ms tolerable time considered by network provider [9].

The drawbacks of path restoration and protections approaches to store large number of rules in SDN switches can be greatly eliminated by SR [3] technology where a logical path of MPTCP subflows can be expressed as a sequence of segments between the ingress and egress network nodes (e.g., a switch/router/link). SR has been recently proposed by the IETF to provide Traffic Engineering (TE) by simplifying control plane where SDN switches no longer need to maintain per-demand routing information or forwarding rules. To overcome the above mentioned limitations, this paper utilizes the MPTCP and SR for reducing failure recovery time, limit the storage requirements and save cost in future 5G softwarized networks.

III. PATH PROTECTION AND DYNAMIC NETWORK RECOVERY USING SEGMENT ROUTING

Path protection and dynamic network recovery from failure is an important aspect of traffic control and management in softwarized 5G networks. Typical failures in SDN/NFV network include: *Link*, *Node* and *Shared Risk Link Group (SRLG) failures* where multiple links can fail together. On failure detection in traditional networks, crankback routing [9]

is normally applied where the traffic is recursively returned back to the source node from which the traffic can be forwarded to the destination node again. However, the crankback routing increases network recovery time because the failure detection window also increases due to the longer path lengths.

Fig. 1 indicates a use case of path protection and link recovery mechanisms using Segment Routing (SR). SR is a source routing approach that aims to provide an advanced packet forwarding mechanism where intermediate nodes (routers/switches) are not required to install flow rules or maintain all steered paths within the network. The details of SR implementation in softwarized networks are available in our previous works [4], [5]. Upon a link failure, the node where the link failure is detected (step 1) reroutes the data packet until the convenient route is found where the packets can be forwarded to their destination point. The novelty of this approach is that, the same data packets of MPTCP flows are tagged first (e.g. with a SR label which contains information on the failed link) and then sent back through the primary path. After receiving the tagged packet (step 2), a reroute node (e.g., S_2) can then respond to the link/node failure and forward the tagged packets (step 3) to their destination node. When the reroute node processes the first tagged packet, a state transition is performed in the OpenFlow switch. A failover table is created and install backup actions where all subsequent packets coming from the source node are forwarded on the reroute node. It is worth mentioning that, when link failures occur in the network, segments can be repaired such that new backup paths are computed for each link in the primary path as shown in Fig. 1. As we demonstrate later in section V, the benefits of this approach are the reduction of path costs, network recovery time and backup path length.

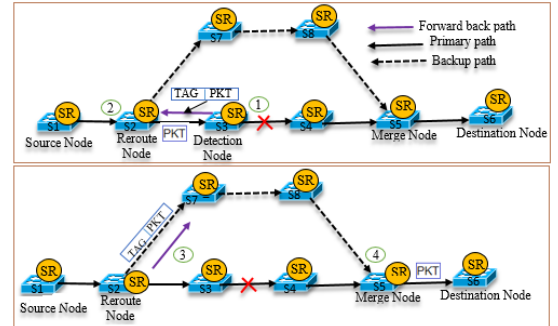


Fig. 1. An example of path protection and link recovery mechanisms

IV. THE PROPOSED MULTIPATH PROTECTIONS AND LINK-RECOVERY MPTCP/SR-ENABLED SDN/NFV-BASED ARCHITECTURE

Fig. 2 shows the proposed architecture for path protections and dynamic link recovery in MPTCP/SR-enabled softwarized 5G network. The proposed architecture consists of forwarding plane and traffic control and management plane. MANO is the NFV management and orchestration framework consisting of three functional blocks: the NFV Orchestrator (NFVO), Virtual Infrastructure Manager (VIM) and VNF Manager.

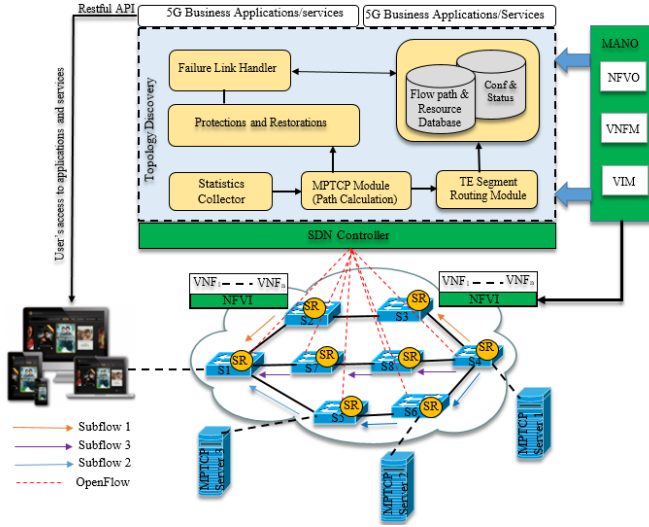


Fig. 2. The proposed path protections and link-recovery MPTCP/SR SDN/NFV-based architecture

A. Data Forwarding Plane

The data plane consists of SDN switches that support SR technology and software/hardware devices (e.g., Virtual Network Functions (VNFs) and OpenFlow switches) interconnected using virtual or wired connections or common wireless radio channels.

B. Traffic Control and Management Plane

The traffic control and management plane includes four modules namely, the *MPTCP-based flow manager module*, the *TE-Segment Routing module*, *Failure Link Handler module* and *Path protections and Restoration module*. These entities were implemented as an extension to POX and OpenDaylight controllers.

1) *Statistics Collector (SC)*: This module collects network information, system events and any network topological changes (e.g., during link or node failure).

2) *MPTCP Module*: As implemented in our previous works [4], [5], this module computes the shortest paths and then performs path allocations to MPTCP subflows. Based on the collected link information, resources (e.g., bandwidth) can be assigned to the calculated paths of MPTCP subflows. Instead of installing these subflow paths in SDN switches as forwarding rules, we use SR approach where the forwarding table of the ingress switch is configured with an ordered list of segments [3]. The ingress switch then adds labels with an ordered list of segments to a packet header and forwards it to its destination point [5], [4].

3) *TE-Segment Routing Module*: To implement the TE-SR module that maps the computed subflow paths to SR paths, we adopt similar approach of the SR assignment algorithm presented in our previous works [5], [4].

4) *Failure Recovery Handler Module*: The failure recovery handler module performs autonomic fault detection and recovery mechanisms for ensuring that any point of failure in the

SDN/NFV can be configured and recovered within a permissible period of time. The NFV management and orchestration (MANO) is responsible for managing and orchestrating virtualized network functions (VNFs). It is important to mention that any changes of users and the running status of the SDN/NFV system can be reported immediately to the SDN controller [5].

5) *Database Module*: The database module stores the configuration parameters, monitoring status reports and different fine-grained resources. All subflow paths with the required end-user's service requirements are maintained in the database. This information enables the SDN controller to use existing paths by querying in the database to retrieve paths that correspond to the previous MPTCP connection. The modules of (1) to (5) are all located in Topology Discover module of the SDN controller as shown in Fig. 2.

6) *The SDN Controller*: In SDN, the controller is solely responsible to detect network failure through Operations Administration, and Maintenance (OAM) tools. SDN uses the OAM tools like Link Layer Discovery Protocol (LLDP) in layer 2 and layer 3 to provide tests for continuity check, connectivity verification, link trace, delay and loss ratio and detect link/node failure. For example, in POX controller, the failure detection is conducted using the POX Topology Discovery module [12]. When a link associated with the Openflow switches is detected as having been removed or failed, a *LinkEvent* message is raised by the Topology Discovery module. The *openflow.discovery* component in POX controller sends LLDP messages out of OpenFlow switches so that it can discover the network topology. For our proposed *PathReLief* algorithm 1, the SDN controller is implemented as an extension to POX and OpenDaylight controllers. Computations of shortest paths of MPTCP subflows is performed using the MPTCP module when the client's request is uploaded to the controller. The controller allocates the MPTCP subflows to transmission paths using the MPTCP module and then maps these subflow paths to SR paths using the TE-SR module (see in Fig 2).

V. MULTIPATH PROTECTIONS AND DYNAMIC LINK RECOVERY IN SOFTWAREZED NETWORKS

A. Network Model

We consider a softwarezied network which is modelled as a directed graph $G = (V, E)$ where V is the set of nodes (switches or VNFs) and E is the set of edges. Each $e \in E$ is associated with a non-negative integer link weight denoted by $W(e)$. Non-fixed parameters (e.g., link bandwidth, packet loss, delay) that indicate link weights in real world network scenarios are used by the SDN controller to calculate the optimal multiple set of h shortest paths from source to destination. The cost of the link, We_{ij} of link (i, j) is defined as the sum of delay (dl_{ij}) and packet loss (pl_{ij}) values of a link multiplied with their scale factors $\alpha, \beta \geq 0$ respectively such that:

$$We_{ij} = \alpha \times dl_{ij} + \beta \times pl_{ij}, \forall (i, j) \in E \quad (1)$$

We denote b_w as the available link bandwidth connecting a pair of SDN/NFV nodes and B_{sf}^k as the required bandwidth of a

subflow sf of an MPTCP connection k . The shortest path of a traffic-flow demand f from the source $s \in V$ to destination $t \in V$ is denoted by $p(s, t)$. We denote a binary variable X_{ij}^p that shows if a certain link is being used or not as defined by equation 2.

$$X_{ij}^p = \begin{cases} 1 & \text{if } p=(i, j) \\ 0 & \text{Otherwise} \end{cases} \quad \forall e \in E \quad (2)$$

In order to ensure that, MPTCP subflows are routed in the network through shortest paths with minimized cost as defined in equation 1, an objective function is formulated as:

$$O_f = \min \sum_{(i,j) \in E} \sum_{k \in K} (W_{e_{ij}} sf_{ij}^k) \quad (3)$$

Where sf_{ij}^k indicates the subflow of a MPTCP connection k routed on link (i, j) . A link may become a bottleneck in real-world network operation scenario when it can not provide the required resources or is likely to be the most used path for any pair of communicating nodes within an SDN/NFV network. We use the *link criticality* l_c parameter introduced earlier in our previous works [4], [5] in order to balance the traffic load and avoid a bottleneck link in future softwarized 5G networks. We define f as a failed link in the network, m as a number of disrupted MPTCP flows and the sf_j^k as the disrupted traffic flow where $1 \leq j \leq m$. In order to achieve a fast recovery after a link failure, the failed link f has to be bypassed using the computed local reroute paths such that:

$$X_{sf,tf}^{p_i} = 0, \forall p \in N \quad (4)$$

Where N is the set of new computed shortest paths after link f has failed and p_i is the i th reroute path, for $1 \leq i \leq n$. n is the number of new computed reroute paths when f fails. The recomputed paths should also have a cycle avoidance constraint defined as

$$\sum_{t \in V, (s,t) \in E} (X_{s,t}^{p_i}) \leq 1, \forall s \in V \quad (5)$$

When the failure occurs in the SDN/NFV network, the bandwidth of the disrupted MPTCP flow assigned to the reroute path p_i can not exceed the available capacity of path p_i .

B. Multipath Protections and Dynamic Link Recovery Algorithm

The proposed algorithm takes the network topology as input and compute link weight using equation 1. The SDN controller calculates the shortest path where MPTCP flows should be transported from s to t . It then performs mapping of MPTCP subflow paths p_{sf} into SR paths. When the client and server establish an MPTCP connection using a three-way handshake, the transmission of subflow is initiated based on user's service requirements. The controller monitors system events and status of network topology. When link (i, j) fails, then it is pruned using algorithm 2 and the network topology is updated. The overall procedure of our proposed approach is summarized in

algorithm 1. The proposed path protection and link recovery mechanism using algorithm 1 does not need crankback routing and the failure detection window and recovery time are minimized.

Algorithm 1: Path Protection and dynamic Recovery of Link Failure (PathReLieF)

input : flow f , $N_topology$ $G = (V, E)$

- 1 Compute link weight based on equation 1, bandwidth;
 - 2 Find all subflow shortest paths $p \in P$ in the network;
 - 3 **foreach** $p \in P_{src \rightarrow dst}$ **do if** $p.used + sf_k.B_{sf}^k < b_{ij}$ **then return** p ;
 - 4 **else** Go to step 3;
 - 5 Perform mapping of subflow paths p_{sf} into SR paths **output:** List of Segment labels SL
 - 6 Save path p with its associated list of SL in DB_p ;
 - 7 Start MPTCP subflow transmission based on user's service requirements
 - 8 The SDN controller monitors system events and status of network topology;
 - 9 **if** link (i, j) fails **then Prune failed link (i, j) using Algorithm 2 **end if** ;**
 - 10 **if** f_{new} is a new flow **then** Query DB_p to find the paths for subflows of f_{new} ;
 - 11 **if** $p_{f_{new}}$ is not in DB_p **then** Go to step 2 and issue path adding request for f_{new} ;
 - 12 **end if**;
 - 13 **end if**;
 - 14 Continue transmission as long as $B_{sf}^k < b_{ij}$;
 - 15 Use the l_c and congestion index to avoid congestion **end for** ;
 - 16 The MPTCP subflow paths are protected and links are recovered
-

Algorithm 2: A Script to Remove a Link from the Network Topology

- 1 *def removeLink(self, node1, node2, port1=None, port2=None, **opts);*
 - 2 **if not opts and self.lopts:** **then** ;
 - 3 self.addPort(node1, node2, port1, port2);
 - 4 $key = tuple(self.sorted([node1, node2]));$
 - 5 $self.link_info[key] = opts;$
 - 6 $self.g.remove_edge(*key);$
 - 7 **return** key
-

VI. PERFORMANCE AND EVALUATION

A. Experimental Testbed & Setup

We use an experimental testbed shown in Fig. 3 which consists of the edge, aggregation and core layers consisting of 8, 4 and 2 SDN switches respectively. Two VMs both running Linux (Ubuntu V16.04 LTS) were installed with the MPTCP

TABLE I
LIST OF USED PARAMETERS

N.Layer	Bandwidth (Mbps)	Delay (ms)	Packet Loss (%)
Edge	3	15	1
Aggregation	2	20	3
Core	3	30	3

v0.92. The Mininet used to model the network redundant links at each level as shown in Fig 3 was installed in another VM. The POX and OpenDaylight controllers were installed in the second VM. The SR module was implemented following the approach we used in our previous works [5]. Table I shows the parameters configured on the network topology shown in Fig. 3.

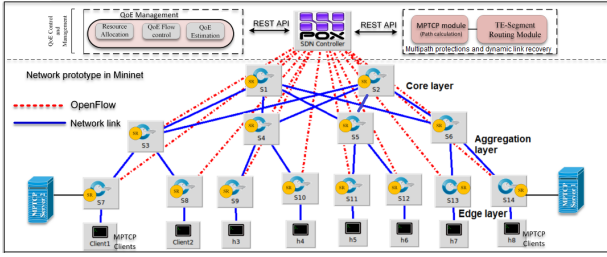


Fig. 3. Experimental Testbed.

B. Experimental Results and Discussion

In order to evaluate the performance of the proposed algorithm, the link between the selected pair of switches were failed by using the pseudocode shown in Algorithm 1. The failed link between each pair of switches was repeated 20 times per pair. The failure recovery time, backup path length and failure localization time were collected to investigate the performance of the proposed algorithm. Failure localization is the process of identifying the exact point where failure has happened. The Failure Localization Time (FLT) is the time taken by the SDN controller to get the link failure information from the POX and OpenDaylight *Topology Discovery* module to the time it finds the exact failure location in the network. The Failure Recovery Time (FRT) is the time taken by the SDN controller to receive the information regarding the exact point of link failure and install the new flow failover tables in SDN switches. The FRT is calculated based on the classical failure recovery model in optical networks presented in [13]. The proposed algorithm 1 for path protection and link recovery was implemented in POX and OpenDaylight controllers respectively. The main idea was to investigate on how the two SDN controllers react to link/node failures in the network in terms of failure recovery and failure localization time.

1) *Failure Recovery Time (FRT)*: Figs. 4 and 5 show the performance comparison of failure recovery time with PathReLief algorithm when it is implemented using the POX and OpenDaylight controllers respectively. The performance of PathReLief is better than that of native approaches used in POX

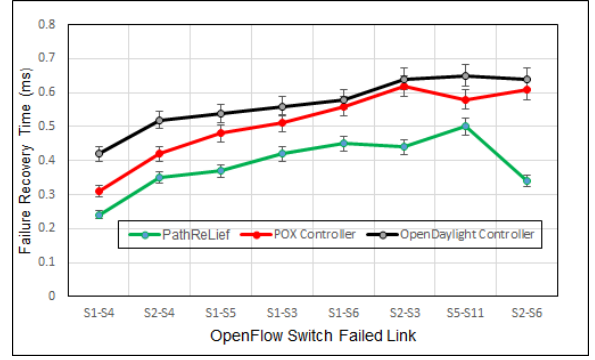


Fig. 4. Performance comparison in terms of failure recovery times using POX controller

and OpenDaylight. This is so because, not only PathReLief uses MPTCP for transmission but it also applies SR technology on every OpenFlow switch which makes the transfer of data from the failure node to the SDN controller fast. It is important to mention that, the SR is disabled while evaluating the failure recovery and localization time using POX and OpenDaylight controllers. It can be observed from Fig. 4, when {S1→S4} fails, the PathReLief takes to only 0.24ms to perform a failure recovery in the network and install failover tables. Note from the network topology shown in Fig. 3, link {S1→S4} connects switches belonging to the core layer and aggregation layer which are closely connected to the SDN controller. This also contributes to the minimization of time required to perform link recovery in the network. The failure recovery time when {S5→S11} fails is the largest (0.5ms, 0.58ms, 0.65ms) for PathReLief, POX and OpenDaylight respectively. This is because link {S5→S11} connects switches belonging to the aggregation layer and the edge layer.

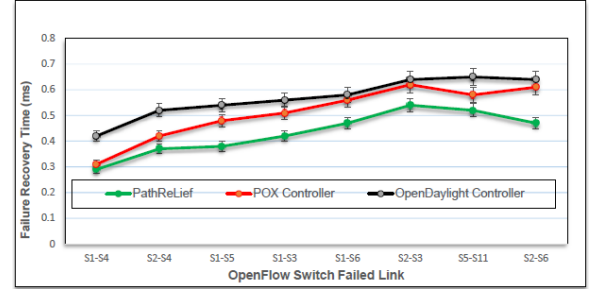


Fig. 5. Performance comparison in terms of failure recovery times using OpenDaylight controller

2) *Failure Localization Time (FLT)*: It can be observed from Fig. 7 that, the PathReLief performs better than the POX and OpenDaylight controllers. For example, when {S1→S4} fails, the failure localization time of PathReLief, native discovery module in POX and OpenDaylight are 48μs, 81μs and 94μs respectively. The time taken for the controller to detect the exact point of a failed link is shorter for PathReLief because {S1→S4} connects the core and the aggregation layer in the fat-tree topology shown in Fig. 3. For failed link {S5→S11},

the failure localization time is $78\mu s$, $85\mu s$ and $96\mu s$ respectively. The localization time is larger because link $\{S5 \rightarrow S11\}$ connects the aggregation layer and the edge layer and therefore making the backup path length to increase.

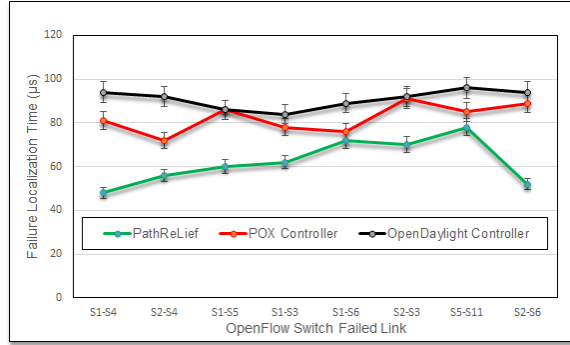


Fig. 6. Performance comparison in terms of failure localization Time

3) *Link Capacity Occupation (LCO)*: The LCO is calculated as the percentage of the total link capacity allocated to the backup paths considered for forwarding the tagged packets. When a failure of a link or a node occurs in the network, some of the available bandwidth are allocated to the backup path depending on the network traffic that are to be sent from the failed node/link to the destination node. The PathReLief shows

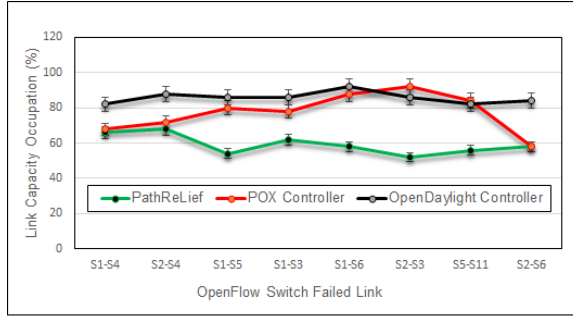


Fig. 7. Performance comparison in terms of link capacity occupation

a minimum link capacity occupation allocated to the backup paths compared to POX and OpenDaylight controller. This is so because MPTCP and SR would enable load balancing between multiple disjointed paths and therefore making the transmission of the tagged packets with minimum bandwidth fast on every link used in the network.

VII. CONCLUSION

In this paper, we propose a path protection and dynamic link- recovery approach using MPTCP and SR to increase survivability, resilience, availability of services in 5G networks. To demonstrate the effectiveness of our proposal, we compare the performance of the proposed algorithm and the conventional topology discovery mechanisms for link/node failures in POX and OpenDaylight controllers. Preliminary results show that, our approach outperforms others used in the commonly used

controllers (i.e., POX and OpenDaylight), in terms of reduced failure recovery time and localization time).

ACKNOWLEDGMENT

The work presented in this paper is fully funded by the European Union in the context of Horizon2020 Research and Innovation Programme under Marie Skłodowska-Curie Innovative Training Networks (MSCA-ITN-2014-ETN), Grant Agreement No.643072, Network QoE-NET.

REFERENCES

- [1] M. Agiwal, A. Roy, , and N. Saxena, "Next Generation 5G Wireless Networks:A Comprehensive Survey," *IEEE Communication Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.
- [2] N. Hyunwoo, D. Calin, and H. Schulzrinne, "Towards Dynamic MPTCP Path Control Using SDN," *IEEE NetSoft Conference and Workshops (NetSoft)*, pp. 286–294, July 2016.
- [3] L. Davoli, L. Veltri, P. L. Ventre, G. Siracusano, and S. Salsano, "Traffic Engineering with Segment Routing:SDN-based Architectural Design and Open Source Implementation," *Fourth European Workshop on Software Defined Networks (EWSN)*, September 2015.
- [4] A. A. Barakabitze, L. Sun, I.-H. Mkwawa, and E. Ifeakor, "A Novel QoE-Centric SDN-based Multipath Routing Approach of Multimedia Services over 5G Networks ,," *IEEE International Conference on Communications*, May 2018.
- [5] A. A. Barakabitze, I.-H. Mkwawa, L. Sun, and E. Ifeakor, "QualitySDN: Improving Video Quality using MPTCP and Segment Routing in SDN/NFV," *IEEE Conference on Network Softwarization*, May 2018.
- [6] I. B. Akbari, D. Krisnandi, R. Wardoyo, and D. Syamsi, "Sub-second path restoration for stateful-based segment routing," *International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, Nov 2018.
- [7] A. Sgambelluri, S. S. S. Anna, A. Giorgetti, F. Cugini, F. Paolucci, and P. Castoldi, "Openflow-based segment protection in ethernet networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 9, pp. 1066–1075, 2013.
- [8] Z. Cheng, Z. Xiaoning, Y. Li, S. Yu, R. Lin, and L. He, "Congestion-aware local reroute for fast failure recovery in software-defined networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 11, pp. 934–944, 2017.
- [9] v. A. Niels L.M, B. J. van Asten, and F. A. Kuipers, "Fast recovery in software-defined networks," *Third European Workshop on Software Defined Networks*, Sept 2014.
- [10] L. Luo, H. Yu, and S. Luo, "Achieving Fast and Lightweight SDN Updates with Segment Routing,," *IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, December 2016.
- [11] Y. Xiong, Y. Li, B. Zhou, R. Wang, and G. N. Rouskas, "Sdn enabled restoration with triggered precomputation in elastic optical inter-datacenter networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 1, pp. 24–34, 2018.
- [12] A. K. Saha, K. Sambyo, and C. Bhunia, "Topology Discovery, Loop Finding and Alternative Path Solution in POX Controller," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2016.
- [13] J. P. Vasseur, M. Pickavet, and P. Demeester, "Network Recovery: Protection and Restoration of Optical, SONETSDH, IP and MPLS: San Francisco, California: Morgan Kaufmann,," 2004.